

CLAIMS:

1. A program storage device, readably by a machine, tangibly embodying programming instructions to perform method steps for constructing a call graph, the

5 programming instructions comprising:

determining for each method M , a set of types S_M of objects that may occur in method M ;

determining for each field F , a set of types S_F of objects that may be stored in field F ;

10 determining the allocation sites inside the body of method M ;

determining the set of directly called methods M' inside the body of method M ;

and

determining the set of virtually called methods M'' inside the body of method M .

15 2. The program storage device according to claim 1, further comprising the programming instructions of:

adding T to S_M for each allocation of type T that occurs in method M .

20 3. The program storage device according to claim 2, further comprising the programming instructions of:

for each direct call to method M' in a body of method M performing the steps of:

adding any type that occurs in S_M and that is a subtype of the type of a parameter of M' to $S_{M'}$; and

25 adding any type that occurs in $S_{M'}$ and that is a subtype of the return type of M' to S_M .

4. The program storage device according to claim 3, further comprising the programming instructions of:

for each virtual call to method M' in the body of method M :

- 5 using S_M , determine each method M'' that may be reached by the dynamic dispatch:
- adding any type that occurs in S_M and that is a subtype of the type of a parameter of M'' to set $S_{M''}$;
- adding any type that occurs in $S_{M'}$ and that is a subtype of the return type of M'' to S_M .
- 10

5. The program storage device according to claim 4, further the programming instructions of:

for each field F read by method M , add any type that occurs in S_F to S_M ; and

5 for each field F with type T written by method M , add any type that occurs in S_M and that is a subtype of T to S_F .

6. The program storage device according to claim 1, further comprising the programming instructions of:

20 using the call graph computed above in a compiler as a basis for performing optimizations such as inlining.

7. The program storage device according to claim 1, further comprising the programming instructions of:

25 using the call graph computed above in a reporting tool to report call graph information to a user.

8. A program storage device, readably by a machine, tangibly embodying instructions to perform method steps for constructing a call graph, the method comprising:

- 5 determining for each method M , only one set of types S_M of objects that may occur in method M ; and
- determining for each field F , only one set of types S_F of objects that may be stored in field F ; and
- determining the allocation sites inside the body of method M ;

10

9. The program storage device according to claim 8, further comprising the steps of:
determining the set of directly called methods M' inside the body of method M ;
and
- 5 determining the set of virtually called methods M'' inside the body of method M .

10. A method for constructing a call graph, the method comprising:
- determining for each method M , a set of types S_M of objects that may occur in method M ;
 - determining for each field F , a set of types S_F of objects that may be stored in field F ;
 - determining the allocation sites inside the body of method M ;
 - determining the set of directly called methods M' inside the body of method M ;
 - and
 - determining the set of virtually called methods M'' inside the body of method M .

11. The method according to claim 10, further comprising:
- adding T to S_M for each allocation of type T that occurs in method M .
12. The method according to claim 11, further comprising:
- for each direct call to method M' in a body of method M performing the steps of:
 - adding any type that occurs in S_M and that is a subtype of the type of a parameter of M' to set $S_{M'}$; and
 - adding any type that occurs in $S_{M'}$ and that is a subtype of the return type of M' to set S_M .

13. The method according to claim 12, further comprising:

for each virtual call to method M' in the body of method M:

using set S_M , determine each method M'' that may be reached by

the dynamic dispatch:

adding any type that occurs in S_M and that is a subtype of the type of a parameter of M'' to set $S_{M''}$;

adding any type that occurs in $S_{M''}$ and that is a subtype of the return type of M'' to set S_M .

14. The method according to claim 13, further comprising:

for each field F read by method M, add any type that occurs in S_F to S_M ; and

for each field F with type T written by method M, add any type that occurs in S_M and that is a subtype of T to set S_F .

15. The method according to claim 10, further comprising the step of:

using the call graph computed above in a compiler as a basis for performing optimizations such as inlining.

16. The method according to claim 10, further comprising the step of:

using the call graph computed above in a reporting tool to report call graph information to a user.

17. A method for constructing a scalable call graph, the method comprising:
determining for each method M , only one set of types S_M of objects that may
occur in method M ; and

5 determining for each field F , only one set of types S_F of objects that may be
stored in field F ; and
determining the allocation sites inside the body of method M ;

18. The method to claim 17, further comprising the steps of:

10 determining the set of directly called methods M' inside the body of method M ;
and

determining the set of virtually called methods M'' inside the body of method M .